

Desktop Science

BY OLAV MARTIN KVERN • ILLUSTRATIONS BY JACK MORTENSBAK



The Right Match. Don't get burned by the wrong graphics file format.

GOOD AND EVIL. RIGHT AND WRONG. YOU KNOW, THE big issues. I have a four-year-old son, which means that I must face, and try to explain, the great moral and ethical issues of human history. Why should you let other kids play with your toys? When, if ever, is it OK to haul off and slug somebody? Why should you wear clothing, wash your hands, or stop jumping on your cousin? If this explaining goes on much longer, I'll end up a philosopher or a lunatic—or both.

Maybe that's why I spend my evenings thinking about more clear-cut issues. Like graphics file formats, which are nothing more than the way your applications write their data to disk, one bit (the fundamental unit of computing—either a one or a zero) at a time. Some formats are definitely right for a particular task, some just as certainly wrong. Some formats are good; others are, let's face it, *evil*.

After fifteen years in the electronic-publishing business (yes, since before it was called “desktop publishing”), I've become familiar, if not intimate, with most

of the major graphics file formats. And I've made most of the mistakes that can be made—choosing formats I trusted only to find, at the last minute before a deadline, that they wouldn't print. If you've been wondering which formats to use, and why, this column is for you.

File-format fundamentals

There are three basic types of graphics file formats.

Bitmap files store pictures as matrices (rows and columns) of squares known as pixels, with each pixel having a particular gray or color value (also known as a gray depth, color depth, or bit depth). Bitmap files are typically created by image-editing programs such as Adobe Photoshop, or by the software you use to run your scanner. TIFF (tag image file format), BMP (Windows bitmap), MacPaint, and PCX (PC Paintbrush) are all examples of bitmap-format graphics files.

Vector files contain sets of instructions for drawing objects—typically geometric shapes such as lines, ellipses, polygons, rectangles, and arcs. The drawing in-

Desktop Science

structions say, “Start this line at this point and draw to that point over there” or “This is a polygon made up of these line segments.” Vector files are typically created by CAD programs such as Autocad. DXF (dynamic exchange format) files are examples of vector-format graphics files. PostScript paths and type, such as those you’d find in an Illustrator EPS (encapsulated PostScript) file, are other examples of vector elements, but they’re usually contained in a metafile.

Metafiles can contain both vector and bitmap graphics, but they don’t have to have both—sometimes you’ll find metafiles that contain only an image, for instance. Macintosh PICT, Adobe Illustrator, EPS, CGM (computer graphics metafile), and WMF (Windows metafile) formats are all examples of metafiles.

There are a lot of different ways to talk about the files saved in these three format types. I usually refer to bitmap files as “images” and vector files as “drawings.” I don’t have a good name for metafiles yet (how about calling them “illustrations” when you place them in another application, and “publications” when they’re complete page layouts on their own).

Note that these are all *interchange* formats—they’re for moving information from one application to another. All programs support their own “native” file format, and many can read or write files in other formats. Some programs can open or import files saved in the native formats of other programs—PageMaker 6.5, for example, can place files saved in Illustrator’s native format. Whenever you move data between programs or save files in formats other than an application’s native format, a process of translation occurs. And, as in

object-positioning information in the file format you’re exporting to? Most vector PICT export filters, for example, round object locations to the nearest whole point. When you export to a vector PICT file from a program capable of greater accuracy (Illustrator, for example), you’ll get rounding errors. Objects will shift on the page.

The reason why

Not all graphics file formats are created equal, and not all are well suited to certain tasks. Why? There’s always a temptation for a parent or a magazine columnist to simply say, “Because I say so.” But I feel that you deserve better. At the same time, a basic explanation of the problems inherent in, say, the Macintosh PICT vector format would consume all of the pages in this magazine. And then there’s WMF, PICT’s Windows counterpart, to think about. There’s just not that much room, so I’ll try to be brief.

Many formats are too device-specific for some uses. The biggest problem with many graphics file formats, in spite of their being designed as interchange formats, is that they make too many assumptions about the system they’ll be viewed on or printed from. We say such formats are *device-specific* because they’re tied to some feature of a particular video-display system or printer (the “device”). Most of these formats assume that files stay on the computer system they’re created on—not a reasonable assumption to make for anyone doing any kind of publishing. On the other hand, PostScript files (including EPS graphics) are practically the definition of a *device-independent* file format—they don’t make many assumptions about the devices on which they’ll be used.

Many bitmap formats incorrectly assume that the color palette of the system they’re on will remain the same. This is particularly true of bitmap-only PICT files, but it’s also true for some varieties of indexed-color TIFFs. Because of this, such graphics can shift or lose colors when you move them from computer to computer.

In addition, most metafile and vector formats—except EPS—assume that the font list of the system they’re created on will remain the same, and refer to fonts by their number (as they appear in the list of fonts at the time the file was created) rather than by their name. This causes huge problems when you move to another system, or even when you install a new font.

If you’re going to use a device-specific graphics file format, you should use it with the devices it’s designed for. WMF, for instance, is a file format that uses commands written in the language of the Windows Graphic Device Interface (or GDI), which is the core system Windows uses to draw objects on screen or print to non-PostScript printers. PICT is based on QuickDraw, the native drawing language of the Macintosh. So if you’ve got a WMF or PICT and you use it exclusively on its native platform (Windows for WMF or



the translation of text from one language to another, the translated version will differ from the original. Sometimes the differences are subtle, sometimes they’re more obvious.

The key to the accuracy of the translation lies in how the native format and the non-native format store information about a graphic—and what the differences are between them. How accurate is the

the Mac for PICT) for on-screen display and non-PostScript printing, you should fare reasonably well.

It all comes down to using the formats with the devices they were designed for. BMP files were intended to be viewed on screen, in Windows—not printed. DXF and CGM files were intended for printing on pen plotters. EPS and TIFF were designed to work well on high-resolution PostScript printers. And GIF (graphics interchange format) and JPEG (Joint Photographics Expert Group) were designed to carry a great deal of image information in the smallest possible package, which makes them ideal for online publishing. In addition, EPS, TIFF, JPEG, and GIF were designed for interchange between different computing environments and platforms—something you can't say of WMF (Windows-only) and PICT (Macintosh-only).

Some programs don't read and write certain formats well. In many cases, the problems associated with a file format aren't caused by the format's specification (the way it was designed)—they're caused by the way the import and export filters read and write the files. For instance, many programs have Windows metafile export filters that change curves into a series of straight lines, even though the current Windows metafile format specification doesn't require that. If you're using high-end, professional graphics applications that are specifically designed to read and write certain formats, you shouldn't have to worry about problems like this.

Some formats just aren't very sophisticated in certain ways. For instance, PICT, CGM, and WMF files use a drawing model that is inferior to that used by PostScript drawing programs such as Illustrator, which offer more accurate, flexible tools for creating EPS files.

Knowing what's good for you

Which file types should you use? If you're creating a document that will never travel across platforms and won't need to print to PostScript devices, you may get perfectly acceptable results from WMF in Windows or PICT on the Macintosh. But if you will be printing to PostScript devices, use EPS for illustrations and use TIFF for images. For artwork you want to include in an online publication, use GIF or JPEG (more on the differences between these two formats below).

I'll explain why, when, and how to use these file types later. In the meantime, you'll generally want to take your Macintosh PICTs, along with your WMF, BMP, PIC, and DXF files, and find some way to convert them into EPS, TIFF, GIF, or JPEG format. If you want to publish something, on paper or online, you'll save yourself time and trouble (and probably money) by doing so. See the chart in the next column for my recommendations on what graphics file format to convert to in various situations.

Here are some good guidelines to follow when you're deciding between the GIF or the JPEG format for online work. If the graphic in question contains primarily

vector artwork or is "synthetic" in nature (contains large areas of flat color or sharp, artificial transitions between colors—screen captures and graphics that started out as vector images are good examples), use GIF. If you're dealing with a "natural" bitmap graphic (images such as scanned photographs that contain subtle, natural transitions between colors), use JPEG.

If you've got a file in this format:	And you want to publish it:	Convert it to:
DXF, vector metafiles, or metafiles with bitmap and vector elements	On paper via PostScript	EPS
PCX, BMP, or bitmap metafiles	On paper via PostScript	TIFF
DXF or vector metafiles	Online	GIF
PCX, BMP, TIFF, or bitmap metafiles	Online	GIF or JPEG

From one format to another

How do you convert files from one format to another? While there are numerous conversion utilities available (such as the excellent DeBabelizer from Equilibrium Technologies, which is now available for both the Macintosh and Windows), most of Adobe's applications can perform file conversions. Photoshop, in particular, can read and save in a large number of bitmap formats, including TIFF, EPS, GIF, and JPEG.

Illustrator can convert Macintosh vector PICTs into editable Illustrator objects. When you place PICT files in Illustrator, any vector artwork in the PICT file is converted into Illustrator paths; any bitmap information is converted to TIFF. Once you see the objects on an Illustrator page, the objects have been converted—all you need to do is save the file as an EPS.

Even PageMaker 6.5 can convert bitmap files from one format to another. (To do so, select a bitmap graphic you've imported, then choose "Graphic..." from the Export submenu of the File menu. In the "Export Graphic" dialog box, select a bitmap format and click "Save.")

Converting vector artwork presents more of a challenge. If you're having trouble converting a drawing from a vector-type file (DXF or CGM, for example) or from a metafile containing vector information, and don't have the application the file originated in (but you do have PageMaker, Illustrator, and the Acrobat Distiller), try this.

1. Place the graphic in PageMaker. Save your document before the next step.
2. Export to Adobe PDF (portable document format): Choose "Adobe PDF" from the Export submenu of the File menu (in PageMaker 6.5) or "Create Adobe PDF" from the File menu (in PageMaker 6.0), and select any options you want before clicking "Ex-

Desktop Science

port..." (PageMaker 6.5) or "Create..." (PageMaker 6.0). For more specific information on exporting to the PDF format, see your PageMaker *User Guide*.

3. Open the PDF with Illustrator and save the document in another format. The process of opening the PDF converts the vector artwork in the PageMaker file to Illustrator paths you can edit. When you're done making any adjustments, save the file in the Illustrator or EPS format.

You can also use this technique to "extract" EPS graphics you have in a PageMaker or FrameMaker publication, but for which you lack the original files.

File-format tips

Now that we've settled on four favored formats, there are a few things you should be aware of.

Save your bitmap images in the right color space.

Use the right type of TIFF for your publication. When you scan line art (black-and-white images), you'll save hard disk space by saving your TIFF files as bilevel (or "Bitmap," in Photoshop), rather than saving them as grayscale or color TIFFs. If you're printing only black ink, your images might as well be saved as grayscale TIFFs, rather than color (again, you'll save space on your hard drive).

If you're printing a PageMaker publication using process inks, you can save your color images as either RGB TIFFs or CMYK TIFFs. If you choose to save your images as RGB TIFFs, PageMaker can perform the color separation as you print (but because it uses color management to do this, it's critical that you select the right source and output device profiles). If you save your images as CMYK TIFFs, you've already created the color separations, and PageMaker simply sends them to your printer. Choose RGB TIFF to minimize the amount of space the images take up on your disk, or choose CMYK TIFF to shorten your printing time.

Open, don't place. Whenever possible, open graphics files rather than importing them. You won't be able to do this in page-layout applications, but Illustrator users should always try to open EPS files, not place

them. Why? Because "nesting" an EPS inside another EPS makes the document much harder to print.

Place, don't paste. When you want to move artwork from one application to another, don't use copying and pasting (though this is OK between Illustrator 6.x and later and Photoshop 3.x and later, since they both support EPS on the Clipboard). Instead, export or save the artwork, then place (or import) it in the destination application.

Why not copy and paste? First, the formats that most applications put on the Clipboard (Windows or Mac) are usually "stripped-down" versions of their file format—you might not get what you need to print the artwork correctly. Next, copying and pasting artwork from one application to another generally doesn't leave you with an externally linked graphic, which is something you might need later to edit the artwork. Finally, you don't always get what you expect from the Clipboard. Sometimes you'll get a bitmap format when you expect a vector format, or vice versa.

Use TIFF clipping paths. When you apply a clipping path to an image in Photoshop, it's easy to assume that you'll have to save the image as an EPS. Given that images saved in EPS files take up more disk space and print more slowly than TIFFs, wouldn't it be nice if you could add the clipping path to a TIFF? If you're creating the TIFF in Photoshop 3.x or later and plan to place the file in PageMaker 6.x, you can—PageMaker is one of the only applications that support clipping paths as part of a TIFF file.

Don't buy what you can't—or shouldn't—use.

You can keep yourself away from trouble by avoiding file formats that don't fit your needs. When you're buying clip art, for example, to use in a newsletter you'll be printing on an imagesetter, you shouldn't even look at packages full of CGM files. Just say no.

On a more philosophical note

Nature versus nurture. Were PICT, or WMF, *born* bad, or were they just brought up that way? Often, graphics file format specifications (how the format was "born") give developers ways to avoid the problems I've cited—but many software developers, especially those who create import and export filters, choose to take the quickest and easiest routes to writing file formats. That's where many problems start. The best way to avoid these problems, as I've said before, is to use formats suitable to the type of work you're doing.

Please feel free to drop me a line care of the magazine, or at olavkvern@aol.com. I'd love to hear from you, and your comments help me make this column better. I can't always reply quickly, but I always read and think about every message. Thanks! ♦

Olav Martin Kvern is a software developer, writer, designer, and illustrator, and is the author of the Real World PageMaker and Real World FreeHand books. His column "Desktop Science" appears in each issue of Adobe Magazine.

